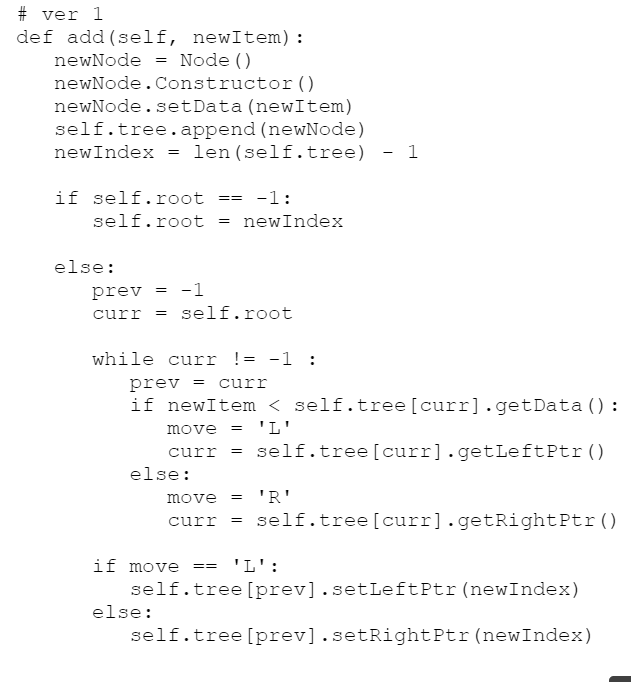
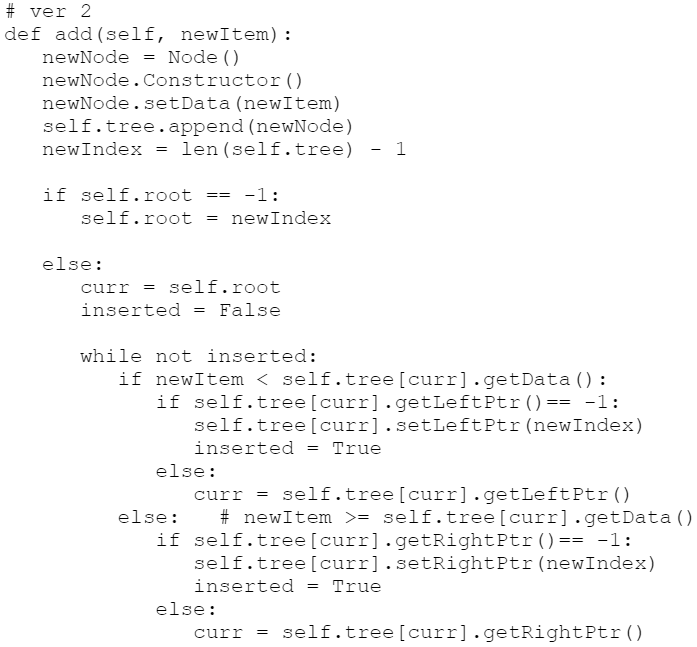
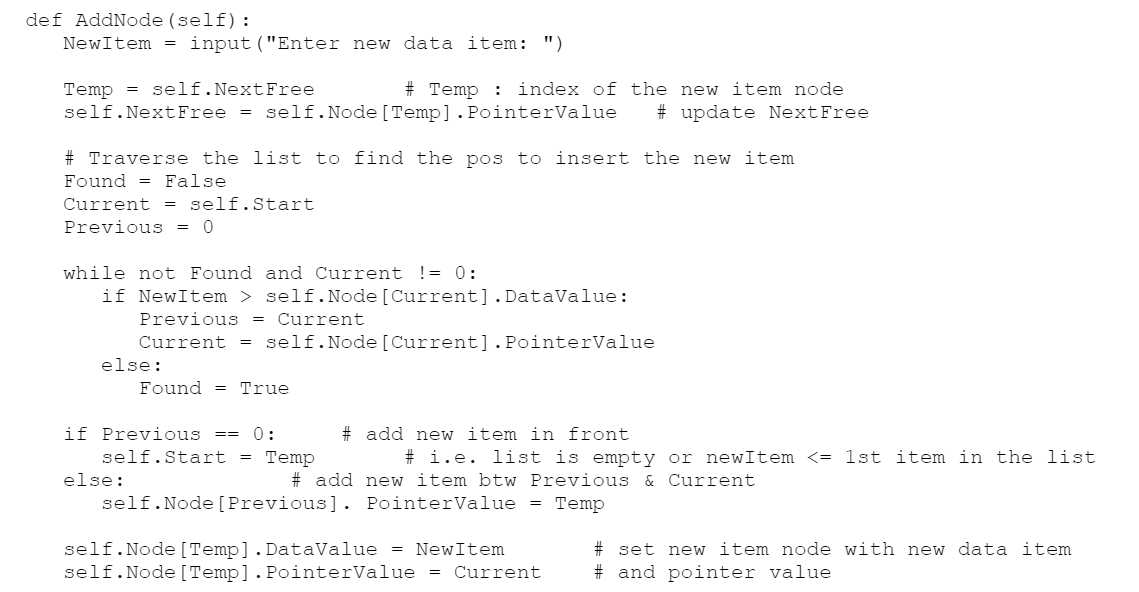
* Follow question paper formatting, ensure every parts is answer
* Do not perform anything in print statement
* Don’t perform multiple actions in a line
* Initalise pointers, Data Value based on what question initialises with
* Pointers should not be initialised to none, initialise to 0/-1
* HAVE BLANK LINES TO SEPARATE CODE
* Adding Items to binary Tree:

1. Initialise new node, then update NextFree



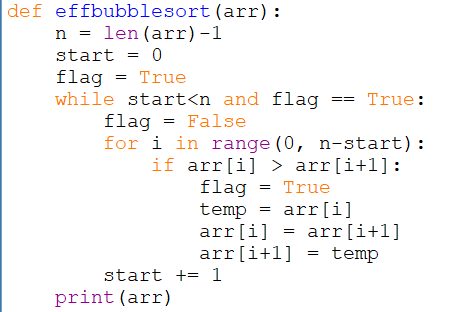


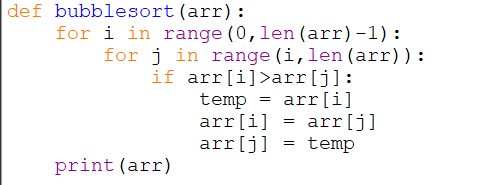
* Use while loop for validation of input
* 3 Test Case:
  + Normal, Invalid, Extreme/Boundary
* Close all open files
* When reading files, .rstrip() first to remove newline at the back of the file
* When asked to describe programme design:
  + Use pointform
  + What type of array (Dimensions?--> [0..3])
  + What functions will be used
  + What datatype
* Don’t need to format nice nice, follow question can le
* Put spaces between screenshots
* When giving users option indicate the possible options:
  + “Enter your choice (1-4): “
* If need to run programme, copy main programme into evidence
* Customised error messages
* Must validate choice if asking users to choose option
* Copy files on CD-Rom to Drive to prevent overwriting
* If given range, should test the range
* Meaningful variable name
* Menu function should only display menu 🡪 Handle choice in main function
* If need find max/min item, set the max/min to be the first item in the list
* Comment for diction
  + IDrecords = {} #id:name
* Instead of constantly saying valid/invalid can XXX “not found”
* Better function names
* Need handle if item not found



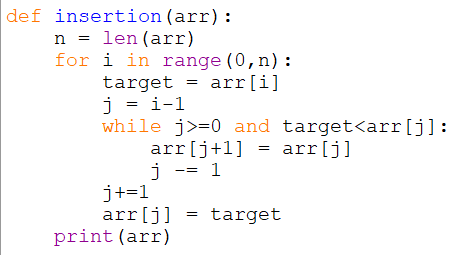
* When designing classes, should have a display function
* Annotation for test cases:
  + Type of data
  + Purpose of test
* Data structure consider: Empty, Full, first/last item

**#bubble sort**

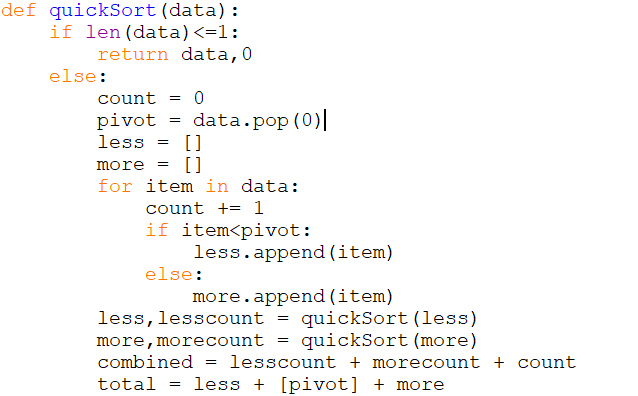




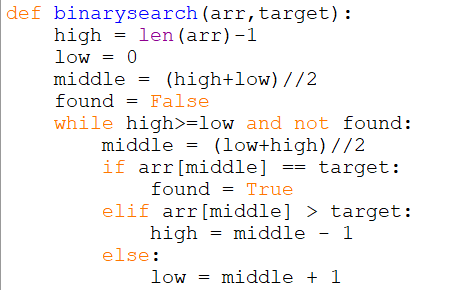
**#insertion sort**



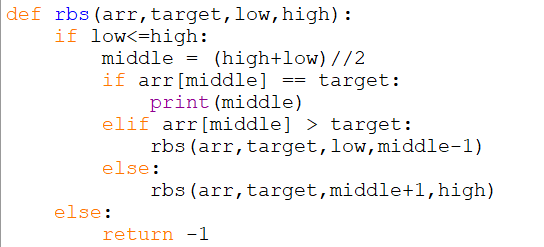
**#quick sort**



**#binary search**



**#recursive binary**



**#Base Conversion**

